# Interpreting the Results: Visualization

- visual/graphical/optical representation of large sets of data:
  - data from experiments or measurements: satellite images, tomography in medicine, microsopy, ...
  - data from simulations:
    - with resolution of time: fluid mechanics, structural mechanics, quantum physics, ...
    - without spatial resolution: vehicle dynamics, optimal control, ...
  - often the only chance to tackle a fast interpretation
  - methods stem from
    - image processing
    - computer graphics
    - virtual and augmented reality
    - ...



# Methods of Image Processing

- image: array of 2D/3D discrete greyscale/colour data
- geometric processing:
  - changes of shapes
  - depend on pixels‘ position, not on their values
  - examples: zooming, rotation, ...
- point-to-point processing:
  - local changes of pixel values
  - depend on current values, but neither on geometry nor on neighbours‘ values
  - examples: addition of constants, contrast stretching, false colours, exponential transformations
- local-to-point processing:
  - local changes of pixels‘ values due to environment

## Methods of Image Processing 2

- local-to-point processing (continued):
  - examples: local weighted averaging (convolution, smoothing), local rank operators (median, sorting), segmentation (thresholds, filtering), edge detection, feature extraction, ...
- ensemble processing:
  - comparison of different images of the same scene
  - examples: detection of motion (military), difference images (medicine, monitoring agricultural activities or vegetation)
- domain processing:
  - complicated modifications of pixel values due to both local and global information
  - examples: Fourier or cosine transform (JPEG), wavelet transform, tomography (CT, NMR, ...)



## Aspects of Computer Graphics

- visualization is based on methods from graphics:
  - crash simulation: representation of cars and the whole scene, perspective view, ...
  - illumination for increasing realism
  - visualizing weather forecast (virtual flights): clouds, fog, ...
  - virtual reality and augmented reality techniques
- geometric modelling:
  - representing 3D objects
- graphical representation / rendering:
  - perspective
  - illumination
  - shading
- stereoscopy (cf. VR)

# Simulation Data Resolved in Time & Space

- given: result (data set) of some numerical simulation

$$f : \mathbb{R}^d \supset \Omega \rightarrow \mathbb{R}^m, \quad m \in \mathbb{N}$$

  f: scalar or vector valued; clear for 2D, but 3D?
- different techniques for the visualization of 3D data:
  - lower-dimensional subdomains:
    - *(ortho-) slices:* the respective quantity is onsidered on some plane only (aligned with coordinate directions or not)
    - *isosurfaces:* draw the surface on which the scalar f or one component of ist components takes some prescribed value t:

$$I(c) = \{(x, y, z) \in \Omega : f(x, y, z) = c\}$$

      interactive creation or modification (change of c, change in time) require efficient algorithms (marching cubes, e.g.)



# Visualizing Simulation Data 2

- visualization of 3D data (continued):
  - lower-dimensional subdomains (continued):
    - *streamlines:* for the visualization of vector fields (m>1); curves with tangent always parallel to local vectors of the given vector field (velocities in CFD, e.g.)
    - *particle tracing:* for the visualization of time-depending vector fields; introduce a virtual particle somewhere and follow its path through the domain according to the vector field
    - *streaklines:* for time-depending vector fields, too; introduce a sequence of virtual particles somewhere and catch their positions at some point of time
    - In the stationary case, all three methods lead to the same lines!
    - furthermore: *streakbands*, *streaktubes* (allow to visualize additional quantities as well as rotation, e.g.)
    - widespread: combinations of the above methods

# Visualizing Simulation Data 3

- visualization of 3D data (continued):
  - colours: use hue or intensity for expressing a scalar quantity
    - use intuition: red/blue for temperature
    - also possible: different clours for different quantities (temperature and concentration, e.g.; limit: about 5)
    - Take care of scale!
  - use of graphical primitives:
    - arrows, balls, ...
    - parameters: colour, size, distance (for speed)
  - use of textures:
    - for indicating material or other chararcteristics (type of soil)
  - use of legends and annotations:
    - very important: which scenario, which quantities, which scale
    - images without legends may be pretty, but not useful



# Component Hierarchy of AVS/Express

- examples of commercial visualization tools:
  - AVS/Express
  - Explorer
- internal representation of data (hopefully able to profit from data structures used for the computations):
- example AVS:
  - component hierarchy (tree structure)
  - field: highest component, whole information
  - mesh: geometric description
    - grid: position of grid points
    - cells: neighbourship relations, connectivity (edges, faces, elements, ...)
  - data: data of the grid (values of f), node data and cell data

# Animation

- tremendous increase in data:
  - one image: some 4 MB, e.g. (1024x1280, 4 B per point)
  - video: some 25-30 images per second!
  - compression necessary (mpg, ...)
- pay attention:
  - smooth transitions (w.r.t. movement, colour, ...)
  - choice of scale:
    - stationary state: too restricted colour-scale at the beginning may suggest convergence even if there are still (small) oscillations
    - increasing differences: now, whole-spectrum colour-state at the beginning may prevent the actual values from being represented
  - if interpolation / inbetweening / frame techniques necessary: don‘t neglect (or even hide) disturbing details!